



# Guía de Despliegue - Plataforma de Cursos

Esta guía te llevará paso a paso para desplegar tu plataforma de cursos en línea en producción.



## Pre-requisitos

Antes de desplegar, asegúrate de tener:

- ☒ Cuenta en [Vercel](https://vercel.com) (<https://vercel.com>) (gratis)
- ☒ Cuenta en [GitHub](https://github.com) (<https://github.com>) (gratis)
- ☒ Base de datos PostgreSQL en producción
- ☒ Cuenta AWS con bucket S3 (para PDFs)



## Opción 1: Despliegue en Vercel (Recomendado)

### Paso 1: Preparar base de datos PostgreSQL

Elige uno de estos servicios (todos tienen tier gratuito):

#### Opción A: Vercel Postgres (Recomendado - Integración nativa)

1. Ve a tu [Dashboard de Vercel](https://vercel.com/dashboard) (<https://vercel.com/dashboard>)
2. Storage → Create Database → Postgres
3. Copia la `DATABASE_URL` que te proporcionan

#### Opción B: Supabase (Generoso tier gratuito)

1. Crea cuenta en [Supabase](https://supabase.com) (<https://supabase.com>)
2. Create New Project
3. Ve a Settings → Database
4. Copia la `Connection String` (modo Pooling)

#### Opción C: Railway (PostgreSQL gratuito)

1. Crea cuenta en [Railway](https://railway.app) (<https://railway.app>)
2. New Project → Provision PostgreSQL
3. Copia la `DATABASE_URL`

#### Opción D: Neon (PostgreSQL Serverless)

1. Crea cuenta en [Neon](https://neon.tech) (<https://neon.tech>)
2. Create Project
3. Copia la `Connection String`

### Paso 2: Fork del repositorio (Opcional)

Si quieres hacer modificaciones:

1. Ve a [https://github.com/davidacaceres/contenedor\\_cursos](https://github.com/davidacaceres/contenedor_cursos)
2. Click en "Fork" (arriba derecha)
3. Esto creará una copia en tu cuenta

## Paso 3: Conectar a Vercel

### 1. Ir a Vercel Dashboard

- Ve a <https://vercel.com/new>

### 2. Importar repositorio

- Click en "Import Git Repository"
- Selecciona `contenedor_cursos` (o tu fork)
- Click en "Import"

### 3. Configurar proyecto

- Framework Preset: **Next.js** (se detecta automáticamente)
- Root Directory: `./` (dejar por defecto)
- Build Command: `yarn build` (automático)
- Output Directory: `.next` (automático)

### 4. Variables de entorno


Click en "Environment Variables" y agrega:

```
env
DATABASE_URL=postgresql://usuario:contraseña@host:5432/database
NEXTAUTH_SECRET=genera-uno-nuevo-con-openssl-rand-base64-32
NEXTAUTH_URL=https://tu-proyecto.vercel.app
AWS_PROFILE=hosted_storage
AWS_REGION=us-west-2
AWS_BUCKET_NAME=tu-bucket
AWS_FOLDER_PREFIX=cursos/
```

#### ⚠ Importante:

- Genera un nuevo `NEXTAUTH_SECRET` para producción
- `NEXTAUTH_URL` debe ser tu URL de Vercel (la verás después del deploy)

### 1. Deploy

- Click en "Deploy"
- Espera 2-3 minutos
-  ¡Tu app está en vivo!

## Paso 4: Configurar la base de datos

Después del primer deploy:

### 1. Instalar Vercel CLI (si no lo tienes)

```
bash
npm i -g vercel
```

### 2. Login

```
bash
vercel login
```

### 3. Vincular el proyecto local

```
bash
cd contenedor_cursos
vercel link
```

#### 4. Ejecutar migraciones de Prisma

```
```bash
# Descargar las variables de entorno de Vercel
vercel env pull .env.production

# Ejecutar migraciones
npx prisma migrate deploy

# Generar cliente de Prisma
npx prisma generate

# Poblar con datos de prueba (opcional)
npx prisma db seed
```
```

### Paso 5: Actualizar NEXTAUTH\_URL

1. Ve a tu proyecto en Vercel Dashboard
2. Settings → Environment Variables
3. Edita `NEXTAUTH_URL` con tu URL real de Vercel
4. Ejemplo: `https://contenedor-cursos.vercel.app`
5. Guarda y redeploy:

```
bash
vercel --prod
```

### Paso 6: Verificar el despliegue

1. Ve a tu URL: `https://tu-proyecto.vercel.app`
2. Prueba el login con las credenciales de seed
3. Crea un curso de prueba
4. Inscríbete como estudiante

✅ ¡Felicidades! Tu plataforma está en producción



## Opción 2: Despliegue en Railway

### Paso 1: Crear cuenta y proyecto

1. Ve a [Railway.app](https://railway.app) (<https://railway.app>)
2. Signup con GitHub
3. New Project → Deploy from GitHub repo
4. Selecciona `contenedor_cursos`

### Paso 2: Agregar PostgreSQL

1. En el mismo proyecto, click en “+ New”
2. Selecciona “Database” → “PostgreSQL”
3. Railway creará automáticamente `DATABASE_URL`

### Paso 3: Configurar variables de entorno

1. Click en tu servicio de Next.js
2. Variables tab

3. Agrega:

```
env
NEXTAUTH_SECRET=tu-secret-aqui
NEXTAUTH_URL=${{RAILWAY_STATIC_URL}}
AWS_REGION=us-west-2
AWS_BUCKET_NAME=tu-bucket
AWS_FOLDER_PREFIX=cursos/
```

## Paso 4: Configurar el build

1. Settings tab
2. Build Command: `yarn install && npx prisma generate && yarn build`
3. Start Command: `yarn start`
4. Port: `3000`

## Paso 5: Deploy

1. Railway desplegará automáticamente
2. Genera dominio público en Settings → Networking
3. Ejecuta migraciones desde tu local:

```
bash
DATABASE_URL="tu-railway-db-url" npx prisma migrate deploy
DATABASE_URL="tu-railway-db-url" npx prisma db seed
```



## Opción 3: Despliegue en Render

### Paso 1: Crear cuenta

1. Ve a [Render.com](https://render.com) (<https://render.com>)
2. Signup con GitHub

### Paso 2: Crear PostgreSQL

1. Dashboard → New → PostgreSQL
2. Name: `cursos-db`
3. Copia la `Internal Database URL`

### Paso 3: Crear Web Service

1. New → Web Service
2. Connect repository: `contenedor_cursos`
3. Configuración:

- **Name:** `plataforma-cursos`
- **Runtime:** Node
- **Build Command:**

```
bash
yarn install && npx prisma generate && yarn build
```

- **Start Command:**

```
bash
yarn start
```

## Paso 4: Variables de entorno

En Environment Variables, agrega:

```
DATABASE_URL=la-url-interna-de-postgresql
NEXTAUTH_SECRET=tu-secret-aqui
NEXTAUTH_URL=https://plataforma-cursos.onrender.com
AWS_REGION=us-west-2
AWS_BUCKET_NAME=tu-bucket
AWS_FOLDER_PREFIX=cursos/
NODE_ENV=production
```

## Paso 5: Migrar base de datos

Antes del primer deploy:

```
DATABASE_URL="tu-render-db-url" npx prisma migrate deploy
DATABASE_URL="tu-render-db-url" npx prisma db seed
```



## Configuración de AWS S3 (Para subida de PDFs)

### Paso 1: Crear bucket S3

1. Ve a [AWS Console](https://console.aws.amazon.com/s3) (<https://console.aws.amazon.com/s3>)
2. Create bucket
3. Nombre: `plataforma-cursos-archivos`
4. Region: `us-west-2`
5. Desmarca "Block all public access" (necesario para URLs públicas)
6. Create bucket

### Paso 2: Configurar CORS

1. Selecciona tu bucket
2. Permissions → CORS
3. Agrega:

```
json
[
  {
    "AllowedHeaders": ["*"],
    "AllowedMethods": ["GET", "PUT", "POST", "DELETE"],
    "AllowedOrigins": ["https://tu-dominio.vercel.app"],
    "ExposeHeaders": []
  }
]
```

### Paso 3: Crear usuario IAM

1. Ve a IAM → Users → Add user
2. Username: `plataforma-cursos-uploader`
3. Attach policies: `AmazonS3FullAccess`
4. Create user

5. Security credentials → Create access key
6. Copia `AWS_ACCESS_KEY_ID` y `AWS_SECRET_ACCESS_KEY`

## Paso 4: Agregar credenciales a Vercel

```
vercel env add AWS_ACCESS_KEY_ID
vercel env add AWS_SECRET_ACCESS_KEY
vercel env add AWS_BUCKET_NAME
vercel env add AWS_REGION
```



## Monitoreo Post-Despliegue

### Vercel Analytics (Gratis)

1. Ve a tu proyecto en Vercel
2. Analytics tab
3. Enable Analytics
4. Ver métricas de rendimiento

### Error Tracking con Sentry (Opcional)

1. Crea cuenta en [Sentry.io](https://sentry.io) (<https://sentry.io>)
2. Create Project → Next.js
3. Instala:

```
bash
```

```
yarn add @sentry/nextjs
```

4. Configura según la guía de Sentry



## Actualizaciones continuas

### Despliegue automático

Con Vercel/Railway/Render, cada push a `main` desplegará automáticamente.

```
git add .
git commit -m "Nueva funcionalidad"
git push origin main
```

### Rollback rápido

Si algo sale mal:

#### En Vercel:

1. Deployments tab
2. Selecciona un deployment anterior
3. Click "Promote to Production"

#### En Railway/Render:













- Similar en sus dashboards respectivos

---

## Checklist final

---

Antes de considerar el deploy completo:

- ☐  App desplegada y accesible
  - ☐  Base de datos conectada y migrada
  - ☐  Datos de seed cargados
  - ☐  Login funciona correctamente
  - ☐  Crear curso funciona
  - ☐  Subir PDF funciona (si configuraste S3)
  - ☐  Estudiante puede inscribirse
  - ☐  Cuestionarios funcionan
  - ☐  Progreso se guarda correctamente
  - ☐  NEXTAUTH\_URL apunta a tu dominio
  - ☐  Variables de entorno configuradas
  - ☐  HTTPS habilitado (automático en Vercel/Railway/Render)
- 

## Troubleshooting

---

### Error: “Invalid `prisma.xxx.findMany()` invocation”

- **Solución:** Ejecuta `npx prisma generate` en producción

### Error: “NEXTAUTH\_URL is not set”

- **Solución:** Verifica que `NEXTAUTH_URL` esté en las variables de entorno

### Error: “Database connection failed”

- **Solución:** Verifica que `DATABASE_URL` sea correcta y la IP esté whitelisted

### PDFs no se suben

- **Solución:** Verifica credenciales AWS y permisos del bucket

### Session no persiste

- **Solución:** Asegúrate que `NEXTAUTH_SECRET` sea el mismo en todos los despliegues
- 

## Recursos adicionales

---

- [Documentación de Vercel](https://vercel.com/docs) (<https://vercel.com/docs>)
  - [Prisma en producción](https://www.prisma.io/docs/guides/deployment) (<https://www.prisma.io/docs/guides/deployment>)
  - [NextAuth.js deployment](https://next-auth.js.org/deployment) (<https://next-auth.js.org/deployment>)
  - [AWS S3 best practices](https://docs.aws.amazon.com/s3/) (<https://docs.aws.amazon.com/s3/>)
-

**¿Necesitas ayuda?** Abre un issue en [GitHub](https://github.com/davidacaceres/contenedor_cursos/issues) ([https://github.com/davidacaceres/contenedor\\_cursos/issues](https://github.com/davidacaceres/contenedor_cursos/issues))